Schneier Chapter 5,6: Advanced and Esoteric Protocols

Chris Brown
University of Rochester Computer Science Department

August 15, 2001

There is of course a lot of material in these chapters and I'm not going to go over it. Following is the outline of these two chapters showing some of the fascinating stuff that is out there. I did want to look at zero knowledge proofs since they seem rather cool.

1 Chapter 5:

Zero Knowledge Proofs

Usually you prove you know something by telling it. This gives away what you know. Here we meet Peggy the prover and Victor the verifier. *Zero-knowledge proofs use one-way functions and interactive protocols to convince V that P knows what she claims: V asks questions that P answers, and if she knows the secret she can always answer correctly. If not she may fail, say, 50 percent of the time. After maybe 10 or 20 questions V is satisfied but he still knows nothing of the secret.

In the Cave of Zero Knowledge, if you know a secret magic password you can pass through the portal from C to D or back. If not, either passage is simply a dead end. P knows secret, wants to prove it to V.

1. V stands at A
2. P goes into cave either to C or D
3. After P is out of sight, V walks to B
4. V shouts to P, asking her either to emerge from the passage to his left or the passage to his right
5. P complies, using the secret word
6. Go thru the shouting and arriving until V is satisfied.

Rather interestingly, if V has a camcorder and records the whole procedure and shows the film of all N trials to Carol, there is no reason she should believe that P knows the magic word. It all could have been rehearsed, maybe they edited bad trials out of the tape, etc. So V can't convince a third party of the proof's validity: his personal interaction is crucial to his belief. You can't learn anything from the proof since to a third party it is totally bogus.

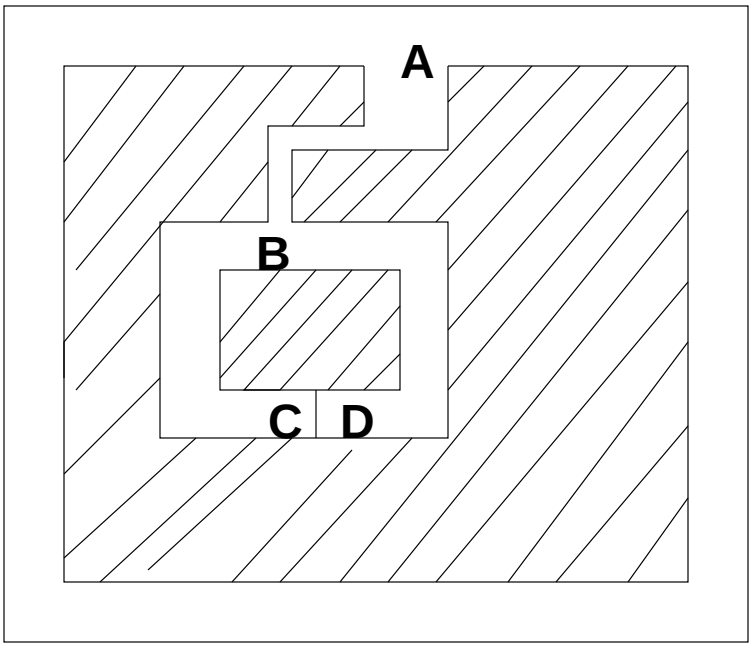

Figure 1: The Cave of Zero Knowledge....Badly redrawn after Schneier

Assume P knows some information that is the solution to a hard problem. The basic zero-knowledge protocol is like this:

1. P uses her information and a random number to transform the hard problem into another hard problem, one that is isomorphic to the original one. She then uses her information and the random number to solve this new instance of the hard problem.
2. Peggy commits to the solution of the new instance, using a bit-commitment protocol.
3. P reveals to V the new instance. V cannot use this new problem to get any information about the original instance or its solution.
4. V asks P either to:
   (a) prove to him that the old and new instances are isomorphic (i.e. two different solutions to two related problems) or
   (b) open the solution she has committed to in step (2) and prove that it is a solution to the new instance.
5. Peggy complies.
6. P and V repeat steps 1 through 5 N times.

As in the cave, a transcript of this interchange could always be faked, so Carol will never be convinced P knows the information by the zero-knowledge proof that convinces V. Technically it isn't trivial to find the right sort of hard problems that support zero-knowledge proof protocols.

As you might imagine, when you see the word "hard problem", you expect something NP complete or NP hard to crop up.

So here's an example using graph isomorphism. Two graphs are isomorphic if their nodes and arcs can be put into 1-1 correspondence. So finding if two graphs G1 and G2 is hard. Suppose Peggy knows the isomorphism mapping and wants to convince Victor she does. Maybe it's represented by a node-adjacency list.

1. P randomly permutes the node labels of G1 to produce the representation of H, which is isomorphic to G1. P thus also knows how to map H to G2. For all other people, finding the isomorphism between G1 and H or between G2 and H is as hard as finding the isomorphism between G1 and G2.
2. Peggy sends H to V.
3. V asks Peggy to either
   (a) prove that H and G1 are isomorphic or
   (b) prove that H and G2 are isomorphic
4. Peggy obliges: she either:
   (a) proves that H and G1 are isomorphic without proving that H and G2 are, or
   (b) proves that H and G are isomorphic without proving that H and G1 are.
5. P and V go through this N times.

P can always create an H that is isomorphic to either G1 or G2. But unless she knows the isomorphism between G1 and G2 she can't get from H to BOTH G1 and G2. So since V can ask her to prove either isomorphism, she has no way to do better than 50 percent unless she actually knows the G1-G2 isomorphism, in which case she succeeds 100 percent of the time. Peggy's new H for each protocol round keeps V from getting information about the relation between G1 and G2.

Finally, V gets a new H and an isomorphism between H and G1 or G2 at each round. Clearly he could work all this out in advance himself and fake the interaction to fool a third party. This is a hallmark of zero-knowledge proofs. The Hamiltonian Cycle is another algorithm that can be used for zkps.

Can you do parallel (not iterative) and noninteractive ZKPs? The latter would be good for convincing Carol. This is cool stuff, using one-way hashes. The upshot is that you can publish data that contains no information about your secret but that can be used to convince anyone of the secret's existence.

Zero Knowledge Proofs of Identity

You can use a zero-knowledge proof to establish that you know your own private key and thus you are who you say you are. This allows you to prove your ID without any physical token. There are lots of cute abuses and ways around this idea, with names like The Chess Grandmaster Problem, The Mafia Fraud, The Terrorist Fraud, The Multiple Identity Fraud, Rending Passports, etc. ...fun stuff!

Blind Signing

Have somebody sign something and keep its content from them.

Simultaneous Contract Signing

Sign a contract simultaneously despite being separated by a distance, with or without an arbitrator, with or without cryptography.

Digital Certified Mail

Bob must sign for receipt of a message from Alice.

Simultaneous Exchange of Secrets

Obvious.

2 Chapter 6:

Secure Elections

Secure multiparty computation

Anonymous message broadcast

Digital Cash